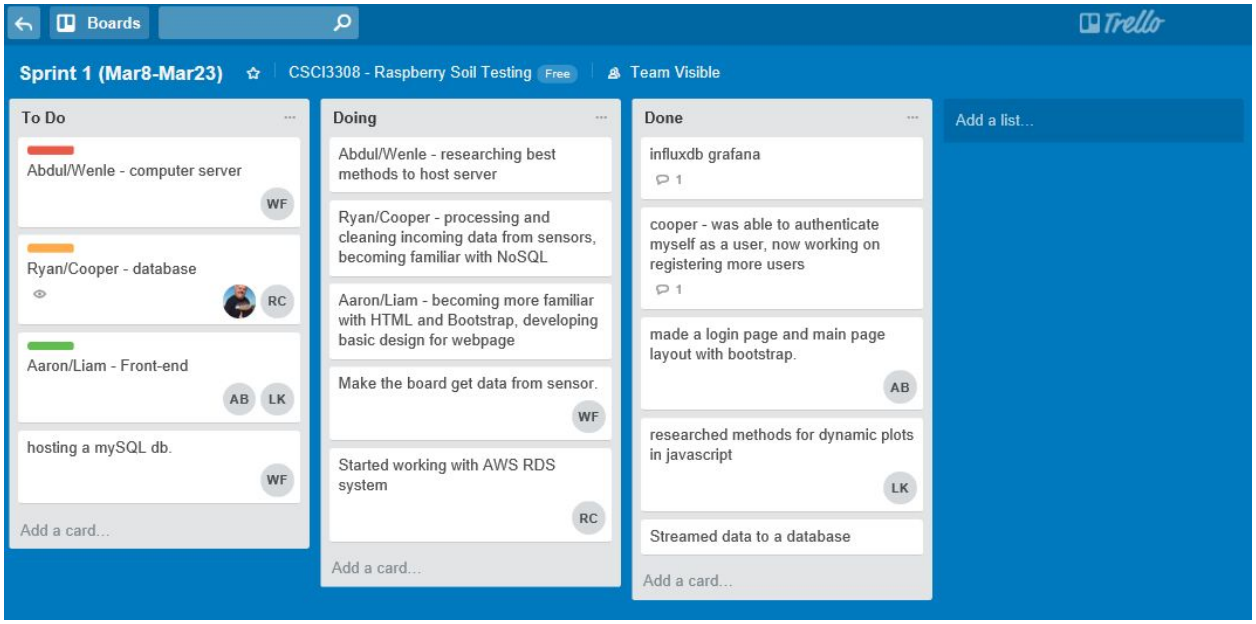
Aaron Barge
Abdullatif AlAbduljaleel
Ryan Close
Wenle Feng
Liam Kolber
Cooper Timmerman

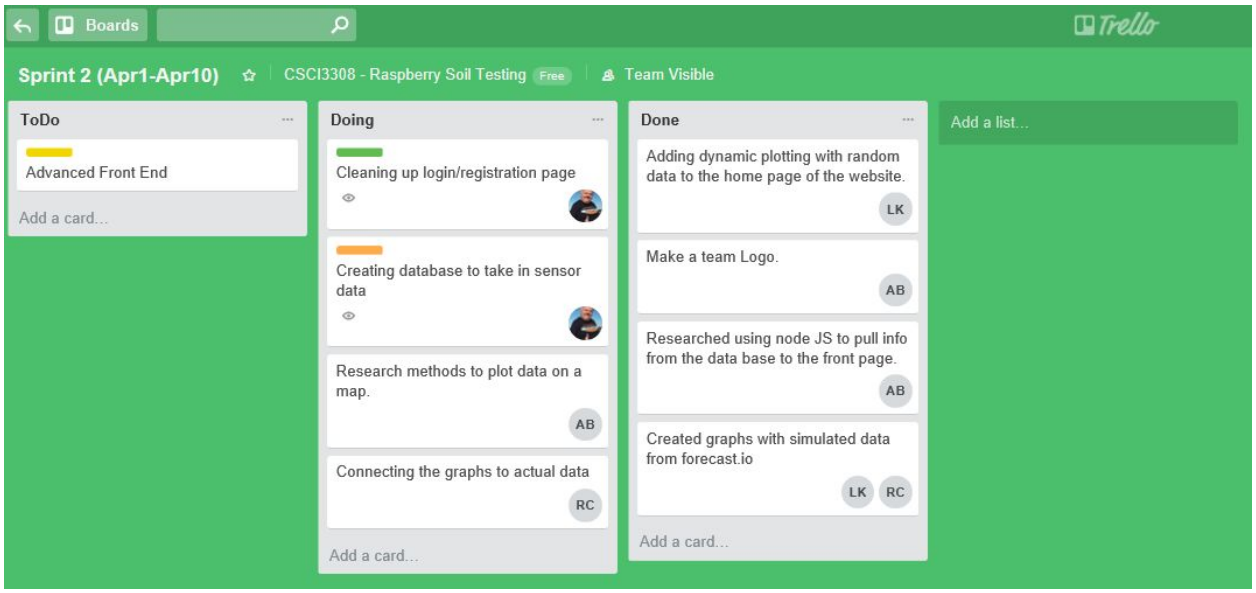
Team Raspberries

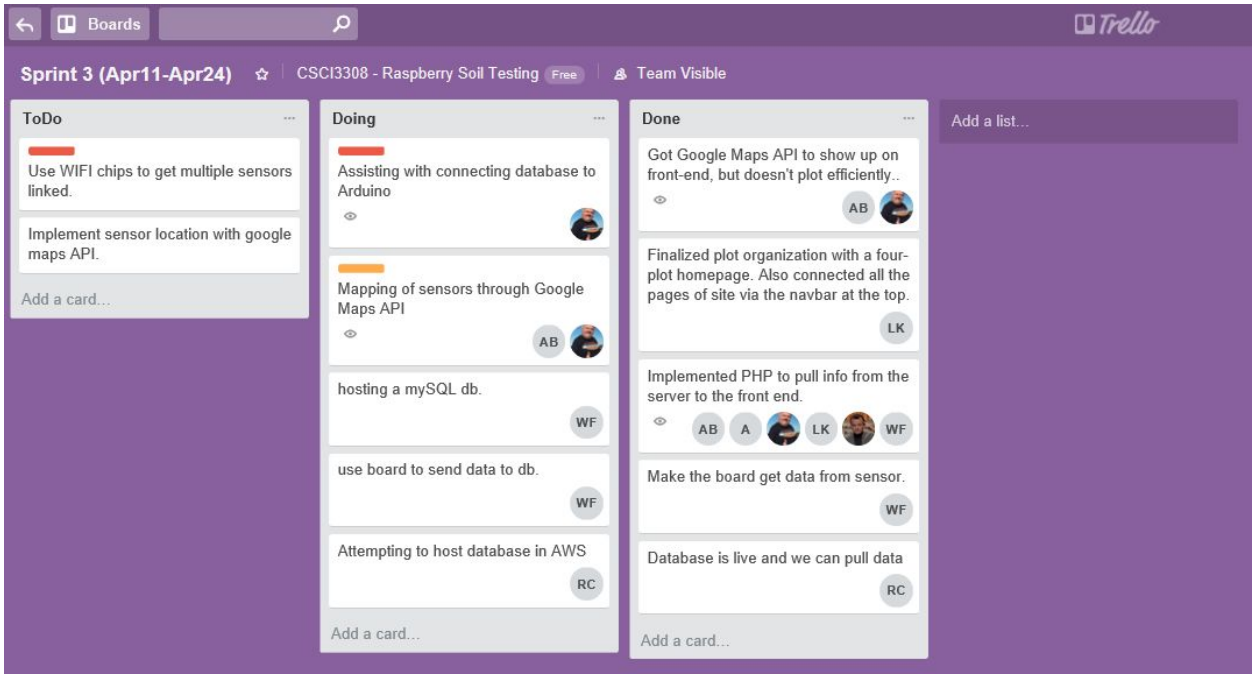
CSCI 3308 - Spring 2018 Semester Project

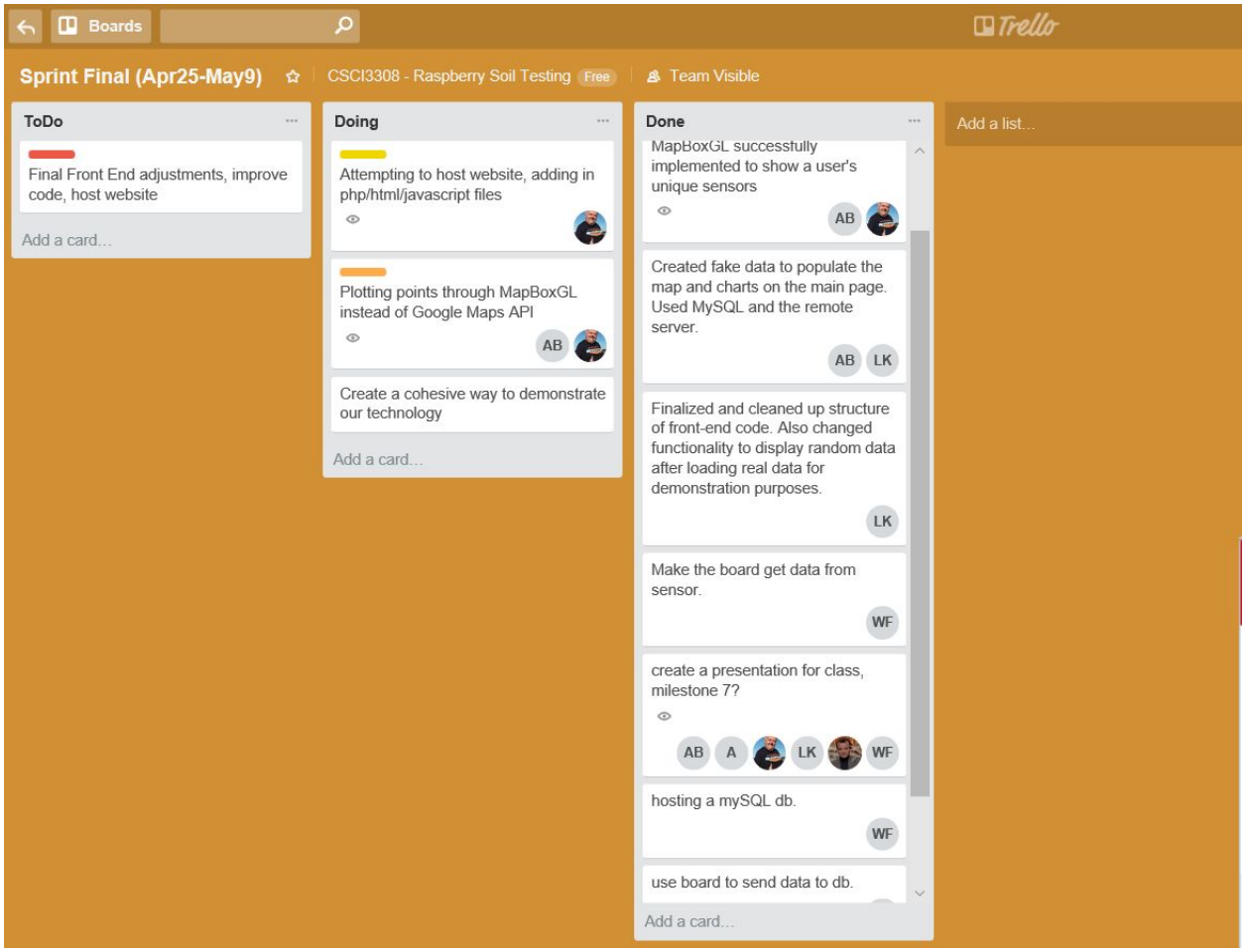
Project Tracker: Trello
https://trello.com/csci3308raspberrysoiltesting

(Screenshots Below)


Boards
Trello
Sprint 1 (Mar8-Mar23)
CSCI3308 - Raspberry Soil Testing
Free
Team Visible
To Do
Abdul/Wenle - computer server
WF
Ryan/Cooper - database
RC
Aaron/Liam - Front-end
AB
LK
hosting a mySQL db.
WF
Add a card...
Doing
Abdul/Wenle - researching best methods to host server
Ryan/Cooper - processing and cleaning incoming data from sensors, becoming familiar with NoSQL
Aaron/Liam - becoming more familiar with HTML and Bootstrap, developing basic design for webpage
Make the board get data from sensor.
WF
Started working with AWS RDS system
RC
Add a card...
Done
influxdb grafana
1
cooper - was able to authenticate myself as a user, now working on registering more users
1
made a login page and main page layout with bootstrap.
AB
researched methods for dynamic plots in javascript
LK
Streamed data to a database
Add a card...
Add a list...


Boards
Trello
Sprint 2 (Apr1-Apr10)
CSCI3308 - Raspberry Soil Testing
Free
Team Visible
ToDo
Advanced Front End
Add a card...
Doing
Cleaning up login/registration page
Creating database to take in sensor data
Research methods to plot data on a map.
AB
Connecting the graphs to actual data
RC
Add a card...
Done
Adding dynamic plotting with random data to the home page of the website.
LK
Make a team Logo.
AB
Researched using node JS to pull info from the data base to the front page.
AB
Created graphs with simulated data from forecast.io
LK
RC
Add a card...
Add a list...


Boards
Trello
Sprint 3 (Apr11-Apr24)
CSCI3308 - Raspberry Soil Testing
Free
Team Visible
ToDo
Use WIFI chips to get multiple sensors linked.
Implement sensor location with google maps API.
Add a card...
Doing
Assisting with connecting database to Arduino
Mapping of sensors through Google Maps API
AB
hosting a mySQL db.
WF
use board to send data to db.
WF
Attempting to host database in AWS
RC
Add a card...
Done
Got Google Maps API to show up on front-end, but doesn't plot efficiently..
AB
Finalized plot organization with a four-plot homepage. Also connected all the pages of site via the navbar at the top.
LK
Implemented PHP to pull info from the server to the front end.
AB A LK WF
Make the board get data from sensor.
WF
Database is live and we can pull data
RC
Add a card...
Add a list...


Boards
Trello
Sprint Final (Apr25-May9)
CSCI3308 - Raspberry Soil Testing
Free
Team Visible
ToDo
Final Front End adjustments, improve code, host website
Add a card...
Doing
Attempting to host website, adding in php/html/javascript files
Plotting points through MapBoxGL instead of Google Maps API
AB
Create a cohesive way to demonstrate our technology
Add a card...
Done
MapBoxGL successfully implemented to show a user's unique sensors
AB
Created fake data to populate the map and charts on the main page. Used MySQL and the remote server.
AB LK
Finalized and cleaned up structure of front-end code. Also changed functionality to display random data after loading real data for demonstration purposes.
LK
Make the board get data from sensor.
WF
create a presentation for class, milestone 7?
AB A LK WF
hosting a mySQL db.
WF
use board to send data to db.
Add a card...
Add a list...

Version Control System: Github
https://github.com/coopdog95/CSCI3308_TeamRaspberries

Deployment:
We recommend you do these steps on a virtual Linux OS if your native machine isn't Linux.

First, make sure an instance of mysql is running:
$service mysql status
If it's not running:
$service mysql start

Once you have a mysql instance running, you can run the "initialize.py" python script, which will set up a database for you with the required schema, and change the config file so it'll have the correct password to your mysql database. It'll ask you to enter your mysql password twice. If you get a mysql error, most likely your mysql instance isn't running (see above).
$python3 initialize.py

Alternatively you could do the above manually, just run the sql script file and change the config.php value of DB_PASSWORD to your SQL password.

Once the database is created on your machine, and the config file is changed to reflect your local database password, you can run the PHP server and test the website.

Running a development PHP server for our website:
Requirements:
-You need to have PHP command line interface installed:
$sudo apt-get install php7.1-cli (for linux machines)
-You need PHP-mysql installed
$sudo apt-get install php-mysql (also linux machines)

Once you have both dependencies, and you've ran the initialize.py file (or made the manual changes), open a terminal, go the soil folder, and run:
$php -S localhost:3000

If you go the url “localhost:3000” it should send you to the login page, and you can navigate from there. You’re now able to register a new account, login with it, and view the Home page where you can see separate charts for your sensor’s temperature and humidity, or navigate to the Map page where we’ve inserted a map that’s pulling latitude/longitude data from a MySQL table.

*Note: if you still get MySQL errors for login/registration, use our own login/hosting method which would be the command:
$ mysql -h den1.mysql6.gear.host -u proj -p
Password: “password.” (that’s all lowercase, no quotes, ending with a period)

Repository Structure:

“soil/javascript/”
-This contains all the front-end javascript sources.
“soil/resources/”
-Any type of media goes here (pictures, logos, etc.)
“soil/socket/”
-This is where we put the socket code for the data streaming.
“maps/”
-This is where our geojson maps are stored, which are used in the actual map html file (viz.html)
“files/”
-This folder contains Milestones, Arduino files, and some additional javascript testing files within its folders, and the rest of the files are self-explanatory.